

ITOS Telemetry Controller

Configuration and Access

Copyright 1999-2006, United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code.

This software and documentation are controlled exports and may only be released to U.S. Citizens and appropriate Permanent Residents in the United States. If you have any questions with respect to this constraint contact the GSFC center export administrator, <Thomas.R.Weisz@nasa.gov>.

This product contains software from the Integrated Test and Operations System (ITOS), a satellite ground data system developed at the Goddard Space Flight Center in Greenbelt MD. See <<http://itos.gsfc.nasa.gov/>> or e-mail <itos@itos.gsfc.nasa.gov> for additional information.

You may use this software for any purpose provided you agree to the following terms and conditions:

1. Redistributions of source code must retain the above copyright notice and this list of conditions.
2. Redistributions in binary form must reproduce the above copyright notice and this list of conditions in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:
This product contains software from the Integrated Test and Operations System (ITOS), a satellite ground data system developed at the Goddard Space Flight Center in Greenbelt MD.

This software is provided ‘‘as is’’ without any warranty of any kind, either express, implied, or statutory, including, but not limited to, any warranty that the software will conform to specification, any implied warranties of merchantability, fitness for a particular purpose, and freedom from infringement and any warranty that the documentation will conform to their program or will be error free.

In no event shall NASA be liable for any damages, including, but not limited to, direct, indirect, special or consequential damages, arising out of, resulting from, or in any way connected with this software, whether or not based upon warranty, contract, tort, or otherwise, whether or not injury was sustained by persons or property or otherwise, and whether or not loss was sustained from or arose out of the results of, or use of, their software or services provided hereunder.

1 ctrlsource.dat

1.1 What is it?

The ctrlsource.dat is an ascii text file that contains definitions for almost all sources available to ITOS. The exception is archive file playbacks. The playback entries are never added to the ctrlsource.dat file.

Each entry in ctrlsource.dat describes how ITOS connects to the source. An entry starts with the name of the source, followed by where to connect, aliases, how frame sorters and unpackers are configured, and how to run programs (when needed). The details of the source entry are detailed later.

1.2 Where does it live?

When ITOS enables telemetry, it looks for the ctrlsource.dat at the following locations (in order):

1. Current Directory
2. Project Home Directory

The Project Home Directory is derived by adding "/home/" to the beginning of the "ITOS_GROUP" environmental variable. For example, when ITOS_GROUP="trace" the Project Home Directory is "/home/trace"

2 What's inside the source configuration file?

The Source Configuration File tells the Telemetry Controller how to connect to data sources. Data sources can be ground stations, programs, computers, anything that provides telemetry data. The source configuration file also can instruct the controller to start a program just before a `frame_sorter` is connected.

Each time the telemetry controller executes a connect command, it reads the Source Configuration File, allowing changes in the source configuration file to take effect immediately.

Each line of the Source Configuration File represents a single source or comment. Comments are denoted by a pound sign "#" as the first character on a line. All fields in a source line must be used, unless otherwise noted. If a field is not needed, "n/a" is used.

2.1 Source Line Defined

Each source is described in the source control file as one line of ASCII text ended with a new line character. Fields are separated by a single space. The ".end" after the *alias list* and *wrapper list* are required.

```
name datatype address mission input_parameters [2ndhdr_appid [OCF_appid]] [packet_archive_annotation]
per_list .end [preprocessor]
```

2.2 Fields

name is the name of the source. This must be unique, and must not contain any keywords, even as substrings. For example, 'rcvudp' and 'pktsin' are not allowed and will cause weird errors.

datatype is the type of data the source provides. The currently may be 'frames' or 'pkts'.

address is the address of the source. It consists of a transport name followed by transport-specific address components. The following transports are supported:

'client_tcp'

or 'tcp', for client-side TCP/IP. This is followed by a set of hostname and port number pairs and other parameters controlling failover and retry. It has the form:

```
client_tcp [cycles] host port [host port [host port]] [retries [interval]]
```

where:

cycles is the number of times ITOS should cycle through the list of host/port pairs trying to establish or re-establish a connection before giving up.

host

port are a hostname and port number to which the ITOS should initiate a connection.

retries is the number of times ITOS should try initiating a connection to each host/port pair in the list.

interval is the time in seconds which ITOS should wait between retries.

'server_tcp'

for server-side TCP/IP. This is followed by a *port* number on which the ITOS should listen for a connection, and, optionally, by an *interval* in seconds that ITOS should wait for a connection request to come in. Once established, if the connection is broken, ITOS will return to listening for another connection for up to the given interval. The default interval is 30 seconds; a zero interval means ITOS should wait forever.

'udp'

for UDP. This is followed by a local *port* number on which the ITOS will read data.

mission is the mission name.

input parameters

gives the type of input data the ITOS can expect to read. This set of fields consists of an input type name followed by three parameters. Accepted type names and their parameter sets are

'ccsds' for CCSDS transfer frames. This is followed by:

frame length

the length of the CCSDS frame from the start of the primary header to the end of the operational control field, if present. It does not include the attached sync marker or any error control fields.

frame version

the expected value of the frame version field. This is zero for Version 1 CCSDS transfer frames.

spacecraft ID

the expected spacecraft identification number.

'aos'

for AOS transfer frames. This is followed by:

frame length

the length of the CCSDS frame from the start of the primary header to the end of the operational control field, if present. It does not include the attached sync marker or any error control fields.

insert zone length

the length of the VCDU insert zone, which usually serves as a secondary header.

spacecraft ID

the expected spacecraft identification number.

<i>ocf flag</i>	if non-zero, indicates that each incoming frame contains an operational control field (OCF); else each frame does not contain an OCF.						
<i>'packet'</i>	for CCSDS packets. This must be followed by <i>'n/a n/a n/a'</i> .						
<i>'spartan'</i>	for SPARTAN spacecraft TDM frames. This must be followed by the length of the telemetry major frame and <i>'n/a n/a'</i> .						
<i>'itp'</i>	for ITP messages containing multiple CCSDS packets. This must be followed by <i>'n/a n/a n/a'</i> .						
<i>'repack'</i>	if the input frames must be aggregated or split to create proper telemetry frames. In some circumstances, DSN stations, for example, will split transfer frames in half, and they need to be reassembled by the ITOS. Repack accomplishes this task. It's parameters are: <table> <tr> <td><i>input length</i></td><td>the length of the input frame.</td></tr> <tr> <td><i>output length</i></td><td>the desired length of the output frame. Input frames will be split or aggregated to create output frames as needed.</td></tr> <tr> <td><i>output sync</i></td><td>the sync marker expected at the head of each output frame. The length of the given sync pattern is used to set an internal idea of the sync size to a number of bytes. The ITOS will check for valid sync at the head of each output frame and attempt to do a byte-wise (not bit-wise) resync if the pattern is missing. If the value of this parameter is zero, no sync checking is performed.</td></tr> </table>	<i>input length</i>	the length of the input frame.	<i>output length</i>	the desired length of the output frame. Input frames will be split or aggregated to create output frames as needed.	<i>output sync</i>	the sync marker expected at the head of each output frame. The length of the given sync pattern is used to set an internal idea of the sync size to a number of bytes. The ITOS will check for valid sync at the head of each output frame and attempt to do a byte-wise (not bit-wise) resync if the pattern is missing. If the value of this parameter is zero, no sync checking is performed.
<i>input length</i>	the length of the input frame.						
<i>output length</i>	the desired length of the output frame. Input frames will be split or aggregated to create output frames as needed.						
<i>output sync</i>	the sync marker expected at the head of each output frame. The length of the given sync pattern is used to set an internal idea of the sync size to a number of bytes. The ITOS will check for valid sync at the head of each output frame and attempt to do a byte-wise (not bit-wise) resync if the pattern is missing. If the value of this parameter is zero, no sync checking is performed.						

2ndhdr appid

is the optional specification of an application ID to be used to forward the CCSDS transfer frame secondary header or VCDU insert zone to the rest of the system. Default value is 2045. Users may define in the ITOS database a packet with this application ID in order to unpack the contents of the frame secondary header and make them available to STOL, displays, etc. If the packet definition contains a time stamp mnemonic, that mnemonic is used to extract the frame time used to set the spacecraft time in telemetry frame archives.

OCF appid

is the optional specification of an application ID to be used to forward the operational control field (OCF) from a CCSDS transfer frame or VCDU to the rest of the system. Default value is 2046. Users may define in the ITOS database a packet with this application ID in order to unpack the contents of the OCF and make them available to STOL, displays, etc. The ITOS inputs to the database define packet 2046 as an OCF containing the command link control word required for COP-1 command verification.

packet archive annotation

has been deprecated and may be removed in a future release. It consisted of the keyword 'paa' followed by an single word (which always is 'anno12').

alias list is a list of alternate names for the source. Multiple aliases are delimited by spaces. One source in the file can be designated as the default by adding the keyword "default" in the alias list.

apid offset The byte offset within an object (i.e. packet) where the Application Id can be found. 'n/a' is placed in this field when the offset resides in the database.

length offset

The byte offset within an object (i.e. packet) where the length of the packet associated with the AppID can be found. 'n/a' is placed in this field when the offset resides in the database.

wrapper list

The wrapper list is zero or more names giving the encapsulation of each input frame or packet. The wrapper may consist of a header, a trailer, or both. Wrappers should be specified in order, from the outermost to the innermost. See the Telemetry & Command Interface Guide for a list of supported wrappers.

preprocessor

is an optional data preprocessor. It consists of two parts: the name of the host on which to execute the subsequent preprocessor command, and the command required to execute the desired preprocessor program, including any arguments. The program is run before ITOS connects to the source, and so can serve as the actual source of data in the case of simulator programs.

Table of Contents

1	ctrlsource.dat	1
1.1	What is it?	1
1.2	Where does it live?	1
2	What's inside the source configuration file? ..	2
2.1	Source Line Defined	2
2.2	Fields	2